UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/798,890	03/12/2004	Andrew Burdass	550-530	4902

23117 7590 06/15/2009

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
FAHERTY, COREY S	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
06/15/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

RECORD OF ORAL HEARING

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS

AND INTERFERENCES

EX PARTE ANDREW BURDASS

Appeal 2009-002029
Application 10/798,890
Technology Center 2100

Oral Hearing Held: May 20, 2009

Before JOSEPH L. DIXON, LANCE LEONARD BARRY, and HOWARD B. BLANKENSHIP, *Administrative Patent Judges.*

APPEARANCES:

ON BEHALF OF THE APPELLANT:

STANLEY C. SPOONER, ESQ.
NIXON & VANDERHYE, PC
901 North Glebe Road, 11TH Floor
Arlington, VA 22203

The above-entitled matter came on for oral hearing on Wednesday, May 20, 2009, at The U.S. Patent and Trademark Office, 600 Dulany Street, Alexandria, Virginia, before Laurie Allen, Notary Public.

JUDGE DIXON: Hello, Mr. Spooner. You have 20 minutes. You may begin when you're ready.

MR. SPOONER: Good morning. I am here on behalf of the assignee of record, ARM Limited, a company in Cambridge, England.

The problem that we're talking -- that the invention solves is set out fairly explicitly in the specification.

If you look at page 4, the -- the opening couple sentences, the first one talks about the normal -- in a computer system, in responding to exception signals, the normal practice is waiting until an instruction boundary or an end or a completion has been reached to initiate the fetching of the exception handling routine; in other words, how you handle that particular problem.

Applicant, recognizing that problem, figured out, hey, why wait? Go ahead, pre-fetch the exception handling routine so it's ready and available if needed, so that you don't have to stall based upon trying to get this information.

So, that's what's in Applicant's claim, a series of three components -- cache memory, instruction pipeline, and an exception controller, and the exception controller operates in a particular manner. It operates in response to an exception signal, as you would expect, but it operates when it's part-way through the execution of the current program instruction, and so, it's not completed yet.

During the middle of that program instruction -- early or late, doesn't make any difference -- this exception signal comes in. At that point, the exception controller triggers a look-up in the cache memory to see if the exception handling program is there.

If it isn't, that's the -- it's as -- if the program is not present within the cache memory, it triggers a cache line fill from the main memory which loads that exception handling routine into the cache memory.

Then it goes on, completes the instruction.

Once the instruction is completed, then it does the last line -- last couple lines of the claim. Upon completion of execution of the current program handling instruction, and if said exception is still current, then the instruction pre-fetch unit fetches it from the cache memory and off it goes. It doesn't have to delay to pull it from the main memory. It's already there in the cache memory, ready to go.

Now, there's a down-side to that problem. If -- and this is discussed in the spec, as well, also on page 4, lines 6 to 12. There are some cases where this pre-fetch, this speculative pre-fetch, is inappropriate, but the problem caused by that is fairly minimal in return to the benefit that is achieved.

Now, I'm not particularly great in the field of computer programming and fault detection and all of that, so I found it helpful to just sort of draw out the time-line of what's going on here.

So, if we just have this one time-line, this is the program instruction, and it is initiated and it goes to completion. Somewhere in here, you have an exception.

JUDGE BLANKENSHIP: Is this background in the brief or in the record somewhere?

MR. SPOONER: No, sir. It just occurred to me in going through all of this, but I mean this is literally what the claim says. So, this language, this timing is specified in the independent claims.

So, an exception occurs sometime during the instruction. I think the claim says part-way through completion of the instruction.

It then looks to see if the exception handling is in the cache. If it is, great. If it isn't in the cache memory, it then calls up this line fill to put it in the cache memory. At that point, this instruction is continuing to crank along here. It occupies a certain amount of time to do all the steps in that instruction.

So, while this is still percolating along, you've already made sure that you have the exception handling routine in the cache memory at that point.

When you get down here at the end, upon completion, if the exception is still valid, at that point, then you fetch the exception handling routine from the cache memory, and off you go. You take care of the exception at that point.

Now, the prior art reference, Birk -- he doesn't do that. He doesn't do any of that. If an exception comes in during his program instruction, Birk aborts the entire thing.

He aborts the instruction, and he goes to a routine that says how do we handle an abort, and presumably in that routine somewhere, he looks up how to find out how to take care of it from main memory or wherever else, but he doesn't say anything about checking to make sure you've got the exception

handling routine in the cache memory so it's readily accessible, and then, when you've completed the instruction, fetch it immediately out of cache and go forward. He doesn't teach any of that.

So, in a nutshell, that's the distinction.

There are some additional problems that the Examiners had.

My particular practice is I like to just try to list the components and the interrelationships and just check to make sure that the Examiner has said, all right, here's a bit in the prior art reference that says this component, that teaches this component; here is an interrelationship between these two components, and this is where it's discussed.

So, I like to have the Examiner identify where these things are in the prior art, so then I can look at them, have the Applicant look at them, and say is he right, is he wrong, whatever. We've had a real difficulty in this particular case in getting the Examiner to commit to any particular structure in the Birk reference.

So, this last clause in the claim, upon completion of execution of said current program instruction -- that's a conditional phrase -- if said exception is still current, another conditional phrase, then something has to occur, an operational phrase.

The instruction pre-fetch unit fetches the exception handling program instruction from the cache memory.

All right.

We've asked where those two conditional phrases and one operational phrase are disclosed anywhere in the Birk patent. He can't find it.

But -- and so, his only answer to pointing out that that's not disclosed anywhere in Birk is the statement in his answer saying, quote, the aborted instruction must be reissued and therefore is completed as required by the claim.

So, he's somehow saying that if the instruction occurs and Birk aborts it, somehow Birk restarts it and then completes it and then does the claimed invention after this eventual completion.

That doesn't happen. There is no disclosure anywhere in Birk that talks about what you do after the instruction is completed.

Birk aborts the instruction when -- when the exception comes up, so that there can't be any of these other things that are specified by the claims.

The Examiner admits, in the Examiner's Answer -- and this is page 8, third paragraph -- he admits that, quote, the "if" limitation of the claim is never met by Birk, meaning that the following limitation, the instruction pre-fetch unit, can never be required.

So, he recognizes that Birk doesn't do any of this. He just aborts, solves the problem, and restarts.

So, that's the first glaring error, I think, in the Appeal Brief that's never explained in the Answer. We again mentioned it in the Reply Brief, but on the record, there is no real answer to that, where it is, how it exists.

The Examiner also admits, in his portion that responds to the Appeal Brief, section C -- he also admits that -- we suggest that the claims do not, upon completion -- the exception will ever be current. Most importantly, the Examiner admits that, in fact -- quote, "in fact, in Birk, it never is," close quote. I think that's a reference to it's never completed.

In Birk, if there's an exception that occurs, it aborts the instruction. It never completes that instruction.

So, as a result, he just then ignores the claim limitations that we're talking about. That's his answer.

So, he doesn't have -- he doesn't show where Birk teaches the claim limitations.

So, the anticipation rejection clearly fails.

Same thing with the obviousness rejection. If he doesn't teach or doesn't show somewhere that the prior art reference teaches the claimed limitations, it can't possibly be sustained. It's not a prima facie case.

We've also argued that it's not a prima facie case even if these elements were disclosed, because there's no rationale or analysis for combining the elements.

He again just says, well, each one of the patents has some advantage or some benefit, and therefore, these benefits would be the reason for combining them, but all patents, by definition, have some benefit.

So, is he saying that every patent is automatically somehow combined? I don't think so. That may be what he's saying, but I don't think that's the test of obviousness under KSR.

Finally, even if there were a prima facie case -- i.e., a listing of all the elements somewhere in the prior art and some reason or rationale for combining them, if there is a prior art reference that teaches away from the combination, that rebuts any prima facie case of obviousness.

Here, we've already discussed Birk teaches go directly to the abort. The exception occurs, you go to the abort. You don't do any of this other

stuff. So, Birk is completely inconsistent with what the Applicant claims here.

Birk teaches away from what we're doing and what we've claimed, so -- and there are other problems with the Examiner's Answer. He ignores some of the sections and that sort of thing.

Applicant is required to put in separate subheadings directed to each of the rejections. Apparently the Examiner is not required to respond to each of them.

So, be that as it may, that's the world we're in, but I'm happy to try to answer any other questions that you might have.

JUDGE DIXON: Any questions?

(No response.)

JUDGE DIXON: Okay. Great.

MR. SPOONER: Have a good day. It's been a long morning.

(Whereupon, the proceedings were concluded on May 20, 2009.)